Technical Document 1334
August 1988

Temporal Knowledge: Recognition and Learning of Time-Based Patterns

C. E. Priebe (NOSC)
D. J. Marchette (NOSC)
C. H. Sung (UCSD)

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152-5000

E. G. SCHWEIZER, CAPT, USN
Commander

R. M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

Information in this report originally was prepared by members of the Architecture and Applied Research Branch (NOSC Code 421) to be presented as a professional paper at the IEEE Applications of Artificial Intelligence Conference held in San Diego, California, during March 1988.

Released by
V. J. Monteleon, Head
Architecture and Applied Research Branch

Under authority of
J. A. Salzmann, Jr., Head
Information Systems Division

LO

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ADA 199911

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NOSC Technical Document 1334	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Ocean Systems Center 6b. OFFICE SYMBOL (if applicable) Code 421	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) San Diego, CA 92152-5000	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Chief of Naval Research 8b. OFFICE SYMBOL (if applicable) ONT	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code) Office of Naval Technology Arlington, VA 22217	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: 62232N PROJECT NO.: RC32S11 TASK NO.: CDB5 AGENCY ACCESSION NO.: ICCDB500

11. TITLE (include Security Classification)
TEMPORAL KNOWLEDGE:
Recognition and Learning of Time-Based Patterns

12. PERSONAL AUTHOR(S)
C. E. Priebe and D. J. Marchette (NOSC) and Dr. C. H. Sung (UCSD)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM TO	August 1988	19

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Gaussian classification
temporal data
recognition patterns
learning patterns

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A self-organizing, distributed, massively parallel network anatomy for the recognition of input stimuli and the learning of temporal patterns is proposed.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL C. E. Priebe	22b. TELEPHONE (include Area Code) (619) 553-4048 22c. OFFICE SYMBOL Code 421

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

Temporal Knowledge: Recognition and Learning of Time-Based Patterns

Dr. Chen-Han Sung*
Carey Priebe**
David Marchette**

*Department of Computer Science and Engineering
University of California at San Diego
La Jolla, CA 92093
and
Department of Mathematics
San Diego State University
San Diego, CA 92182

**Architecture and Applied Research Branch
Naval Ocean Systems Center
San Diego, CA 92152

ABSTRACT

A self-organizing, distributed, massively parallel network anatomy for the recognition of input stimuli and the learning of temporal patterns is proposed. The network adapts itself to recognize individual incoming events in the first, or static, subsystem. These recognized events, received by the system over time, are simultaneously categorized as specific sequences by the temporal subsystem. Seperate attentional mechanisms allow for the recognition of events with a low signal-to-noise ratio while simultaneously allowing attention in the temporal subsystem to be focused only on sequences that meet some minimum length criterion. The static subsystem is based on the adaptive resonance paradigm of S. Grossberg. The temporal subsystem, a gaussian classifier, processes the static information produced by the first subsystem. These gaussian classifications represent the statistics of the temporal data and use a scheme of moving mean and moving covariance to update the classes. Via supervised learning these self-developed classes are then combined into an overall probability estimate using a bayesian probability scheme. The temporal subsystem calculates a gaussian distance from its multi-dimensional input, and hence has the important ability to predict the presence of a sequence prior to receiving the entire sequence as input. Each subsystem has autonomy in the updating of its memory traces, and the information stored is independent of the simultaneous action of the other subsystem. The stability of the system as a whole is guaranteed by the relative independence of the subsystems.

The Requirement for Temporal Knowledge Processing

The ability to understand one's environment, an essential property in the elusive search for intelligence, is not governed by static pattern recognition alone. The order in which events occur can be even more important than the events themselves, and an intelligent system, whether it be a mouse or a robot, must be able to detect and understand this ordering. Thus the dimension of time allows access to a wealth of information about the current environment, past events, and expectations about the future. An ability to incorporate time into information processing is necessary for abilities such as recognition of sequences of events, understanding cause and effect, making predictions, planning, et cetera.

The ability to recognize sequences is essential for many tasks, most notably those involved with audition and vision. A sequence may consist of a stream of phonemes, typed letters or frames from a movie. Once the initial preprocessing has been done and the individual members of the sequence have been recognized, the task is shifted. The processing is then concerned with determining which of the known sequences are represented by the input. Since the answer may depend on the context in which the input has been received, the system must return all the sequences that the input might represent with a confidence rating indicating the quality of match between the input and the known sequences.

Consider a stream of letters. The task is to recognize the words that are being typed. Here it is important to recognize the order in which the letters appear and the words these sequences of letters might represent. If the typist is poor, a certain amount of error will appear in the sample and this must be dealt with. Also, a subsequence may be a legitimate word, and the system should recognize this in order to allow a more sophisticated system to deal with the ambiguities. For instance, suppose the space bar is broken and the typist is prone to error. With the following input the system might respond as shown:

Input: itakeisssuewitjthet
Output: I it take I is issue sue wit with the that

The system might produce a confidence level for each word. After the final "t" the last two words, "the" and "that", might have roughly the same confidence, signifying that each is nearly correct and are valid responses. With the added information of word boundaries that the space bar would educe, the number of spurious words detected would be significantly reduced.

To process this stream of letters the system must be able to represent order information and work on imperfect exemplars. In a more complicated situation, e.g. spoken language recognition, the duration of the constituents of the sequence and the spacing between them may become important. The system must be able to incorporate this information, and ideally it should learn the sequences and adapt to the environment in which it will be operating.

The system proposed is a multi-layer architecture which uses two distinctly different network paradigms. The network adapts itself to recognize individual incoming events in the first, or static, subsystem. This subsystem utilizes the Adaptive Resonance paradigm outlined by Carpenter and Grossberg (3). These recognized events, received by the system over time, are simultaneously categorized as specific sequences by the second, or temporal, subsystem. Seperate attentional mechanisms allow for the detection of events with a relatively low signal-to-noise ratio while at the same time requiring the sequences to meet some minimum length criterion. The temporal subsystem, a gaussian classifier, processes the static information using a combination of temporal decay and moving mean and covariance (4) to obtain a representation for the statistics of the input patterns and update the categorizations. These self-developed categories are then combined via supervised learning to produce the final output, an array of the confidences that the system is seeing each of the learned sequences.

Proposed Anatomy

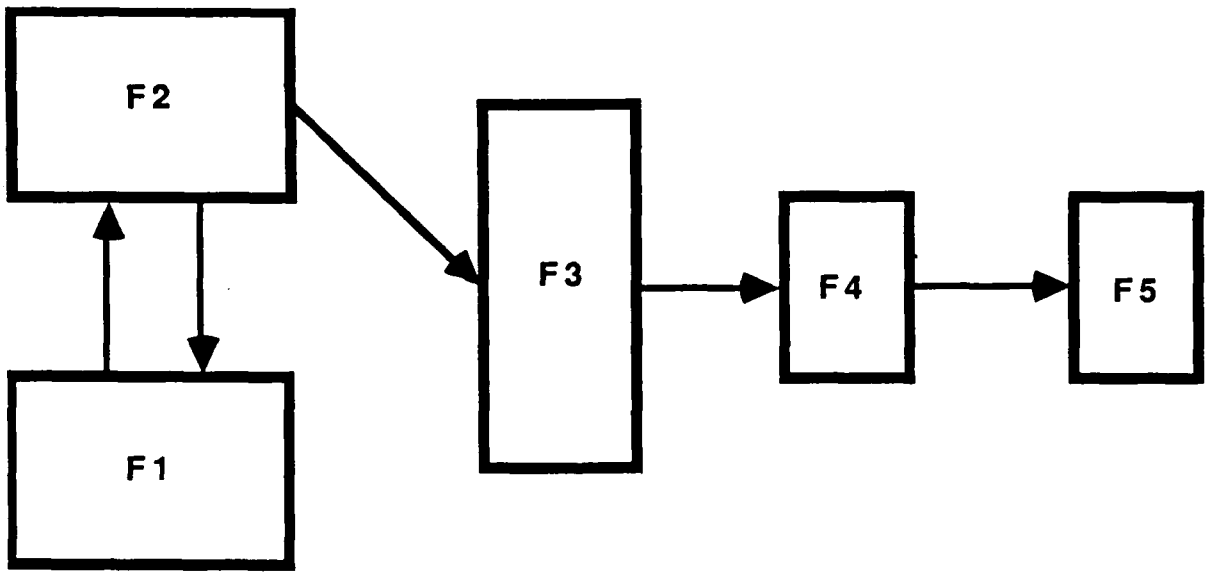

FIGURE 1

Temporal information processing system anatomy. Fields F1 and F2 constitute the static subsystem, an adaptive resonance system. Field F3 contains information based on the classifications made by F2, after decay and shunting. The F4 field, together with the connections from F3 to F4, consist of the gaussian classification partitions. Field F5 is used to combine self-organized sequence classes via supervised learning. This allows for the incorporation of expert rules.

Figure 1 shows the proposed temporal system anatomy. The system is broken into two distinct subsystems: static and temporal. The subsystems interact via specific connections from F2 to F3. Self-organization is an important element of the system, and it is this criterion that guides the design. The system must be able to adapt to changes in the environment, and to recognize novel inputs and new sequences while at the same time watching for known patterns. In other words, learning should never be turned off, although some capability to prime the system externally should exist. Each subsystem performs learning, the updating of connection weights and activation function parameters, independently. The knowledge stored in these memory traces is derived from both external inputs and information coming from the other subsystems. The learning, however, is done in a closed subsystem without being affected by the concurrent processing being performed in other sections of the system.

Static Subsystem

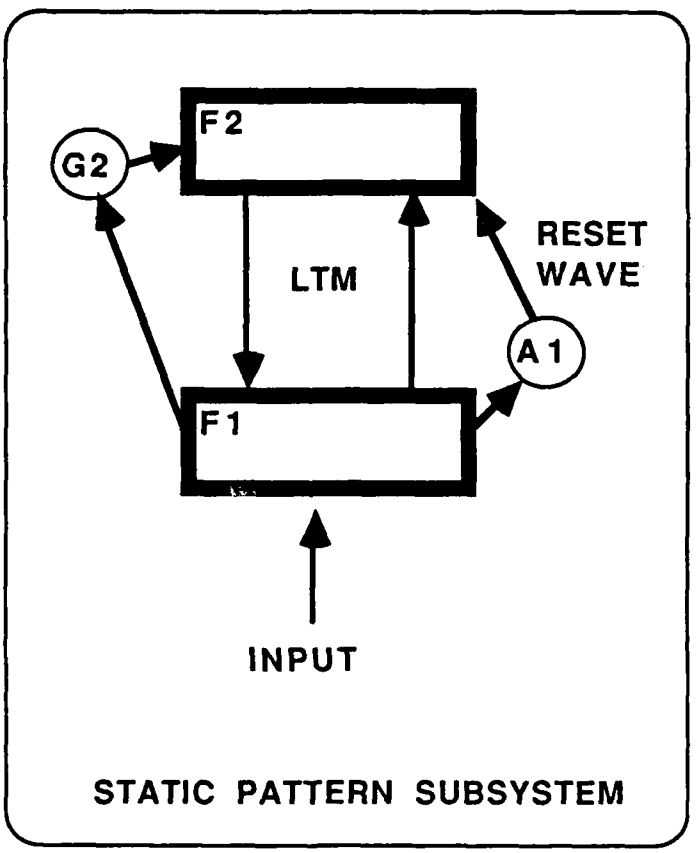

FIGURE 2

Static subsystem: Adaptive Resonance Theory

The requirements of the static subsystem, or S1, are that of a straight- forward pattern classification system. Inputs to the system need to be processed and classified according to some criteria. Some attentional functionality is useful, but in this anatomy a third subsystem concerned with context and expectancy would relay attentional information to the F1 field.

S1 is implemented as the simplest form of an adaptive resonance system, with minor adaptations. Long-term memory (LTM) traces between fields F1 and F2 contain learned knowledge on the classification of incoming events. This classification will, in the simplest case, take the form of a single event-node in the F2 field passing information to the temporal subsystem and field F3. In the general case, it might be advantageous to pass multiple real-valued signals from F2 to F3. However, the equilibration and stability properties of adaptive resonance theory under such a multiple-node classification scheme are insufficient. In addition, the static subsystem need only perform a single-node classification, as the temporal subsystem will accept single events and integrate them with respect to time. It is quite possible that under the more general case of multiple-node static classification, an intermediate subsystem would be necessary to perform precisely the transformation to single-node events prior to input into the temporal subsystem.

The transfer of information from F2 to F3 is performed only after the S1 has reached resonance. At this point the F2 field has encoded the event information from the input. This subsystem is self-organizing, in that the event categories are created by the system based on some error criterion, or vigilance, and a threshold in the gain-control node G2. The vigilance level, represented by the activation threshold of the A1 node, regulates the amount of noise allowed in a pattern, as well as the relative coarseness of the individual categories. The G2 threshold level, when used in combination with the 2/3 matching rule (3), determines the amount of activity necessary in field F1 to activate field F2. This allows the system to ignore a certain amount of noise in the system when unaccompanied by signal. This feature avoids the pitfall of having the temporal processing fail to correctly classify important sequences due to attention being paid to non-events.

Temporal Subsystem

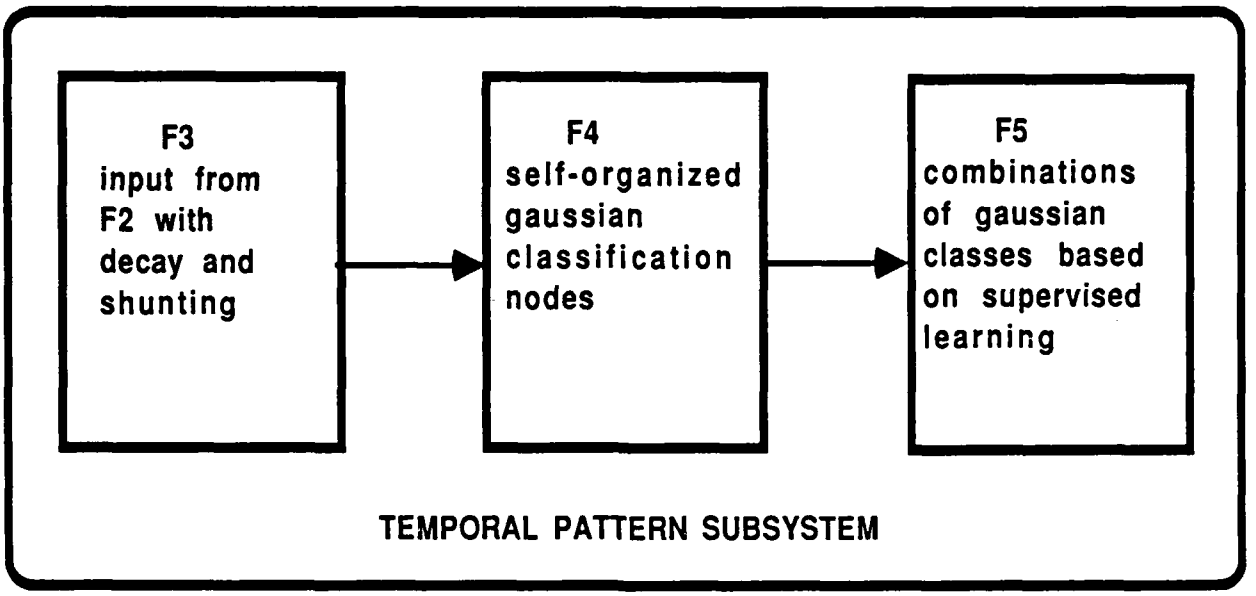

FIGURE 3

Temporal subsystem consisting of three fields. The gaussian classes are represented in field F4.

Upon receiving categorization results from the static subsystem, the temporal subsystem, S2, attempts to classify the spatial representation of the sequence received thus far into an existing temporal pattern category. Failing this, the subsystem creates a new category for the current sequence.

The F3 field receives input from the F2 field once the static system has reached resonance and has settled upon a classification category. This category is then passed to F3. The F3 field experiences a decay factor, providing for the ordering of its nodal activations based on their time of input (5). Some shunting based on the attention to be given to a particular category could also be provided (6).

The F4 field utilizes a gaussian classification scheme to achieve an unsupervised partitioning of the input space. Each node consists of a multidimensional gaussian activation function in which the mean and covariance matrix adapt to the data. The subsystem then learns the statistics of the data by representing the data as a sum of multi- dimensional normal distributions.

In one extreme, in which each input represents a distinct class and each node learns just one data point, this subsystem produces a Voronoi classifier (1). The Voronoi classifier for a set of points is the optimal nearest neighbor classifier for the points. In this case, the subsystem is nothing more than a nearest neighbor classifier, returning the gaussian distance from each of the stored points.

In the other extreme, in which all the points are classified by a single node, the subsystem fits a normal distribution to the data. In this case the subsystem would compute the mean and covariance matrix for the inputs and its output would be a normal distribution with those parameters.

A large class of distributions can be approximated by a mixture of gaussians. Therefore the temporal subsystem approximates the distribution of its input by a collection of gaussians at the F4 level. This allows for a distributed system in that it uses many nodes to represent the distribution of the input, and since the domain of the gaussians is infinite, there is a degree of redundancy in this representation. The system degrades gracefully under nodal failures, yielding the fail-safe property that is desirable in many applications.

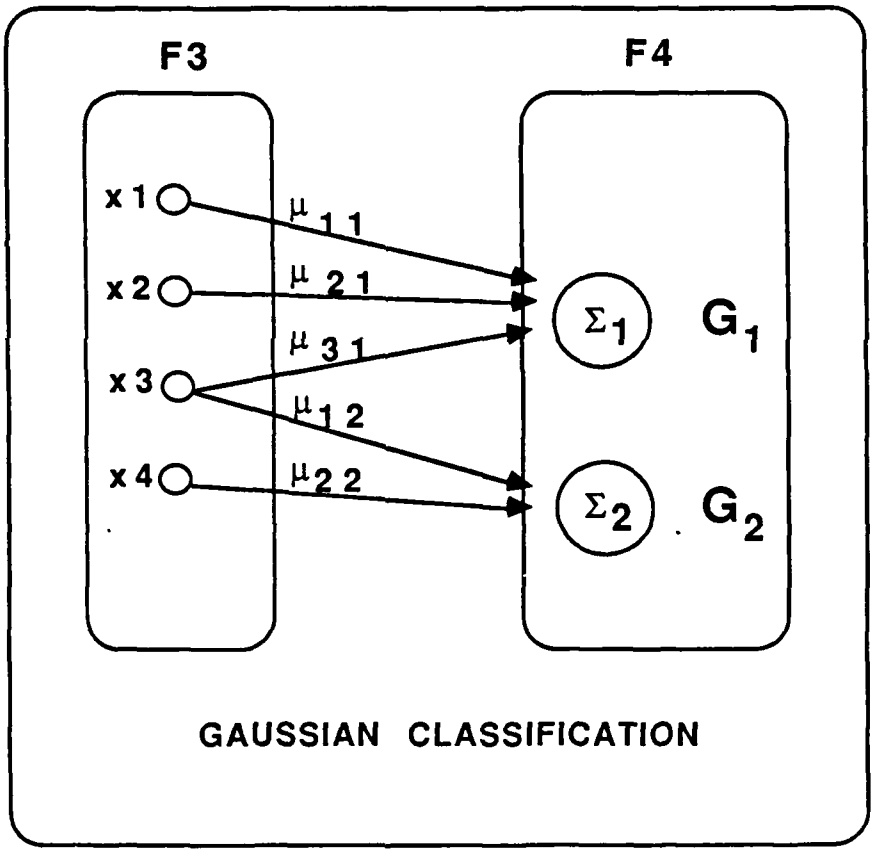

FIGURE 4

Representation of gaussians:

The incoming weights contain the mean, μ , while the covariance, Σ , is contained within the gaussian node itself. The dimensionality of each individual gaussian is determined by the number of non-zero incoming weights. In this case, G_1 has dimension 3 while G_2 has dimension 2.

The individual gaussians are represented by the nodes in the F4 field. Each gaussian may be of a different dimensionality. The covariance for a particular gaussian is represented by the activation function of the node, while the mean can be represented in the connections from F3 to F4 (see figure 4). The gaussians are presented with the n-dimensional input from field F3 having n nodes and, in parallel, compute their respective activation values (gaussian distances) from this input. The activation value for these gaussian nodes in F4 can be obtained via the following formula:

$$a_j(\underline{x}) = \frac{\exp\left(-0.5[(\underline{x} - \underline{\mu})^t \cdot \Sigma^{-1} \cdot (\underline{x} - \underline{\mu})] \right)}{(2\pi)^{(d/2)} * |\Sigma|^{(1/2)}} .$$

Here $a_j(\underline{x})$ is the activation value of the j^{th} gaussian node when presented with vector input $\underline{x}$. Σ is the covariance matrix for gaussian j, while $\underline{\mu}$ is the vector-valued mean. d is the dimensionality of this particular gaussian and is equated with the number of non-zero input connections to node j. Since the components of the mean $\underline{\mu}$ are represented as a node's input weights, these weights can be thought of as shifting the origin for the node. A gaussian function is then applied to the inputs of the node, as opposed to a sigmoid (see figure 5).

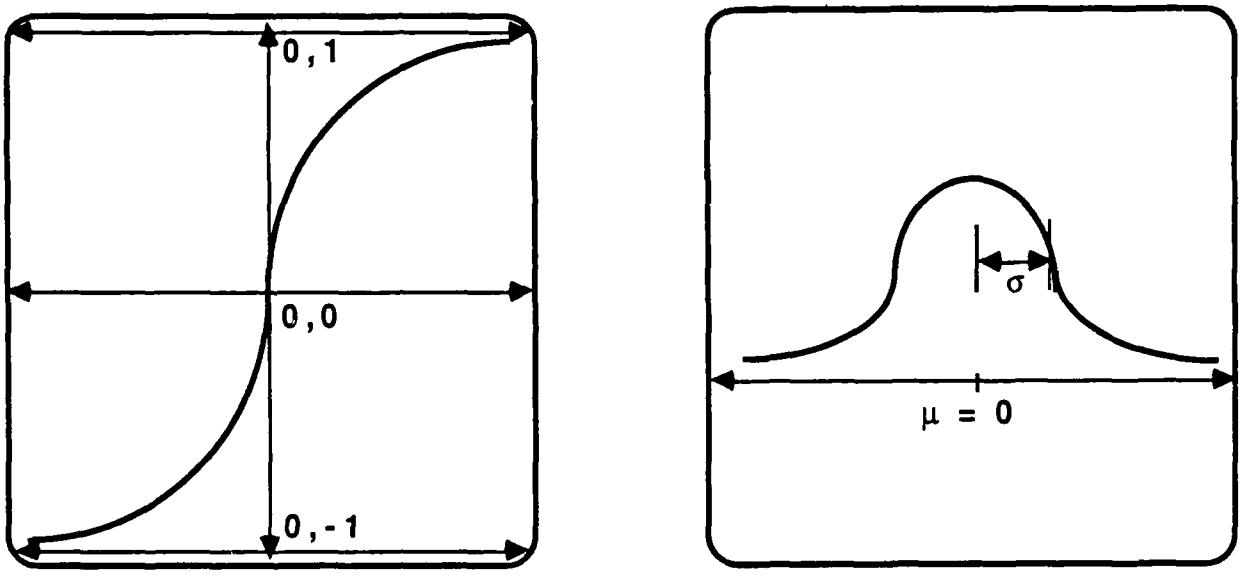

FIGURE 5

Activation functions:
(a) shows the conventional sigmoid. (b) shows the gaussian activation function used in field F4. The incoming weights shift the origin for the function, and the variance is a plastic, or learned, parameter.

Only those nodes whose activation values reach some threshold defined by that node's activation function (usually the value of the gaussian at one standard deviation from the mean) are considered to be a likely category for the current input and are updated. This updating consists of moving the mean and variance of the gaussian based on the current input and some measure of the total number of inputs to the gaussian thus far. For the updating of the mean, we have

$$\mu(j+1) = \mu(j) + [1/(j+1)][I(j+1) - \mu(j)]$$

where $\mu(j)$ is the (one-dimensional) mean after input j and $I(j+1)$ is the $j+1^{st}$ input to be categorized in this gaussian. The updating of the covariance is similar. Here

$$S_{xy}(j+1) = S_{xy}(j) + [j/(j+1)] * A_{xy}(j+1) \quad ,$$

with

$$A_{xy}(j+1) = [(x(j+1) - \mu_x(j)) * (y(j+1) - \mu_y(j))] \quad .$$

Then

$$\Sigma_{xy}(j+1) = \frac{1}{j} S_{xy}(j+1) \quad ,$$

where we are calculating $\Sigma_{xy}(j+1)$, the x,y component of the covariance matrix Σ after the $j+1^{st}$ input clustered in this gaussian. $x(j)$ and $y(j)$ are the x and y component of the input vector after the j^{th} input, and μ_x, μ_y are components of the mean.

Independent of this learning procedure, the values across the F4 field represent the relative likelihoods that the current input belongs to a particular class. In the unsupervised case, this is the solution, and the system, having been presented with a given sequence, will be able to recognize that sequence, along with similar sequences that have missing or additional features. In fact, the activation value of an F4 node is a measure of how close a given input is to previously learned sequences. If we allow a particular gaussian to process on only those inputs that are a part of its make-up, i.e. allow gaussians to only process within their dimensionality, we have a fundamental way to ensure that extra or spurious input, as would be found in a noisy environment, do not affect our recognition of learned sequences.

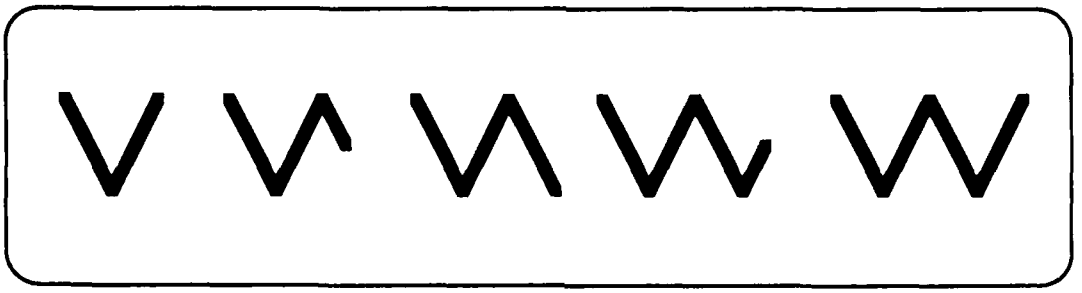

FIGURE 6

Pattern Drift:
A recognition system must be able to notice the drift, and continue to recognize patterns within the context presented. At the same time, the system's categorization should not be fatally affected by the input drift.

This statistical learning scheme allows for a measure of control over the drift of categories. This can be seen by considering the example in figure 6, in which an initial block **V** is slowly deformed into a **W**. The input pattern gradually becomes degraded. In a strict adaptive resonance system, this can force the category corresponding to the original pattern to drift toward the most degraded pattern. This necessitates the eventual creation of a new category for **V** the next time it is presented to the system. In addition, in a temporal setting the old **V** category, which now represents **W**, will still be a part of previously learned sequences containing **V**. These sequences must also be relearned, incorporating the new **V** category. Until this relearning has occurred, the system will consider a sequence containing the new **V** as different from those containing the original **V**, even though the sequences may actually be identical. The gaussian classification scheme will instead represent the statistics of these patterns as they have been presented. Therefore, a mixture of categories will exist that represents the average of the patterns that have been coalesced. While the original **V** node will be modified as the category drifts, this modification is restrained and will only allow a small amount of drift, as defined by the covariance. More significant drift will cause new gaussians to be created. This is a result of the infinite support of the gaussian activation functions. Each gaussian node returns its distance from the input, regardless of the degree of fit between the node and the input pattern. This implies the ability to learn the tendencies of the input data in a non-chaotic way. In addition, nodes that by themselves represent **V** and **W** continue to exist, and need not be recreated later.

A consequence of this scheme is that it allows the system to recognize when events are no longer being detected in the environment. Each node retains a time tag indicating the last update time, allowing the system to recycle nodes that no longer are being used by the system. This is important in an implementation, since the constraints of a finite number of nodes, combined with a shifting environment, can cause the system to saturate.

The activation values of these F4 nodes are also used in a supervised learning setting. If information exists as to the meaning of the sequences, this information is utilized by field F5 in combining the self-organized classifications into actual event categories. We set the connections between F4 and F5 such that, for an event category (F5 node) that consists of more than one F4 class, the corresponding F5 node receives input from each relevant F4 node (See figure 7). In this example, a reading system will need a gaussian class for both upper case **A** and lower case **a**, as well as for **B** and **b**. The stimulus **A** is clearly different from **a**, and needs to be categorized seperately. It is unreasonable to expect these vastly different stimuli to be categorized together, at least in this first level. However, in preparing voice output, these four categories must be coalesced into two categories, **A** and **B**, where each category may contain several phonemes. This is done via supervised learning. The system is taught that, in this case, **A** and **a** should both indicate a final F5 output of **A** as in figure 7. This affords the system the modularity to process the input patterns independent of their eventual meaning within the context indicated by the environment. This corresponds to the actual processing that must be done by intelligent systems. The vision system must classify the pattern input for **A** and **a** seperately, and the eventual combination of the two patterns into a single conceptual category is dependent on the context and function of the system.

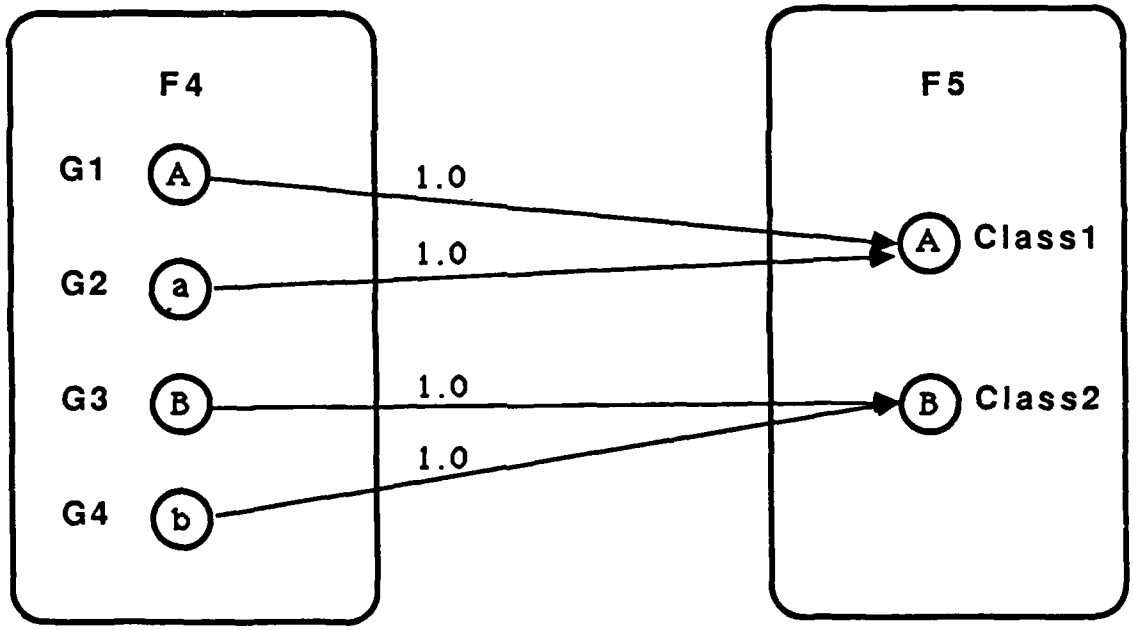

FIGURE 7

Supervised gaussian classification combination:
Gaussians G1 and G2 are combined into a single class in F5, as are G3 and G4. Thus both representations for **A** can be mapped to the same high-level entity.

This can also be generalized to using known a priori probabilities for the

events and combining the F4 nodes in a bayesian probability sense. Consider the morpheme sequence **"tu"**. Suppose it is known, through either expert knowledge or observations by the system, i.e. statistics regarding the inputs seen thus far, that the meaning indicated by **"tu"** corresponds to **"to"** 50% of the time, **"two"** 20% of the time, and **"too"** the remaining 30% of the time. The connections between F4 and F5 can represent this knowledge, as in figure 8. The meaning for the indicated sequence has a priori probabilities representing these ratios. The context, or surrounding words, is then used in conjunction with these a priori probabilities to determine the posterior probability for each potential meaning. This bayesian combination scheme allows for the incorporation of current information (context) with a priori knowledge (prior statistics). In this way expectation is added to the system, based on these prior statistics. This allows the addition of expert knowledge to the system, allowing a degree of interaction between the system and the developer or user.

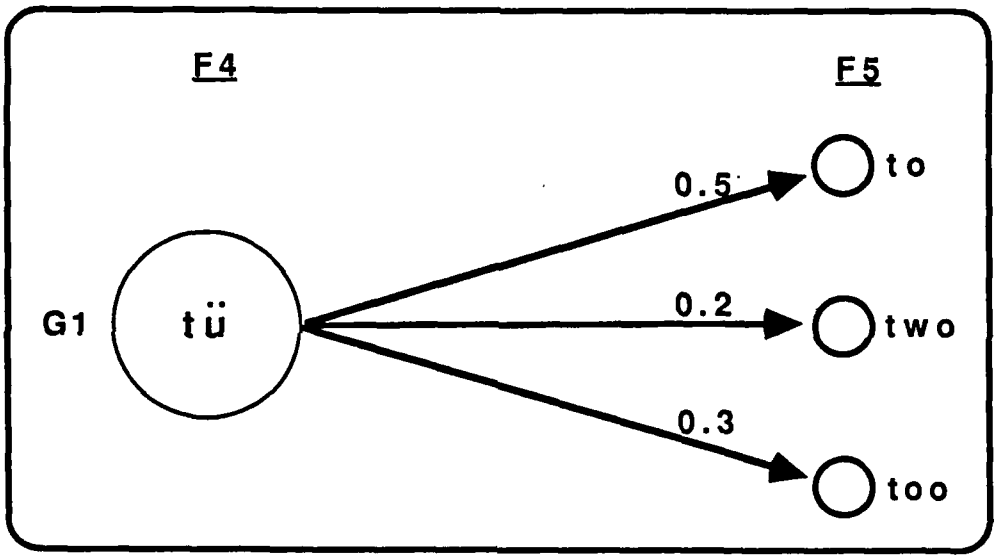

FIGURE 8

Supervised gaussian classification combination:
Gaussians are processed in a probabilistic sense, based on system feedback or expert knowledge.

Summary

The need for more complete temporal knowledge processing has been clear to researchers in artificial intelligence and neural network theory for more than twenty years. Work in rule based systems has failed to yield a satisfactory approach. In addition, much of the neural network research in the area has been devoted to a simple transformation of temporal data into a spatial pattern. Although this is the approach which lends itself to early small-scale success, it is necessary to use the incoming temporal data in a way that preserves the knowledge inherent in this data. The different aspects of temporal information -- short-term, medium-term and long-term

context -- indicate a seperate approach to extracting the knowledge for each aspect. The system proposed herein performs processing on incoming data without first depriving it of much of the information of importance. It learns the statistics of its input and the system adapts to a changing environment. Although specific architectures may vary, the component concepts described above will allow a system to utilize the temporal information available to more fully understand its environment.

Acknowledgements

Special thanks to Dr. Dennis Healy, Deptartment of Mathematics, Dartmouth College, and Dr. Roger Johnson, Naval Ocean Systems Center. Work at Naval Ocean Systems Center was supported in part by the C^3I Data Fusion Task in the Office of Naval Technology Command Systems Technology Block Program and the Joint Directors of Laboratories' Parallel Inference Engine project.

References

1. Aggarwal, Chazelle & Guibas, Parallel Computational Geometry (1985), *Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science*, 468-477.

2. Calvert & Young, (1974) **Classification, Estimation and Pattern Recognition** , American Elsevier Publishing Company, New York.

3. Carpenter & Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine (1987), *ComputerVision, Graphics, and Image Processing*, (37) 54-115.

4. Chan, Golub, & LeVeque, Algorithms for computing the sample variance: analysis and recommendations (1983), *The American Statistician*, (37) 242-247.

5 Grossberg (1986) in **Pattern Recognition by Humans and Machines,** Vol 1, eds. Schwab & Nusbaum, Academic Press, 216-226.

6. Grossberg & Stone, Neural dynamics of attention switching and temporal-order information in short-term memory (1986), *Memory and Cognition*, 14 (6), 451-468.

7. Sklansky, ed. (1973), **Pattern Recognition,** Dowden, Hutchinson & Ross, Inc.

8. Tank & Hopfield, Neural computation by concentrating information in time (1987), *Proc. Natl. Acad. Sci.* USA 84, 1896-1900.

9. Watanabe (1985), **Pattern Recognition: Human and Mechanical**, John Wiley & Sons, Inc.